



Escola Politècnica Superior
d'Enginyeria de Vilanova i la Geltrú

UNIVERSITAT POLITÈCNICA DE CATALUNYA

SISTEMA DE MESURES DE DEFORMACIONS DE GALGUES EXTENSIONOMÈTRIQUES

Aplicació per a pràctiques docents

PDUSADGE

Kenneth Colell Solé

kenneth.colell@upc.edu

Germán Morillo Velázquez

german.morillo@upc.edu

Resum:

En aquest projecte s'ha dissenyat i produït un sistema de mesures d'extensiomètria a mida, amb una sèrie de característiques definides per l'usuari. Es tracta d'un sistema de mesures mitjançant galgues extensiomètriques complet per a les pràctiques d'extensiomètria i elasticitat, disposa de 16 bits de resolució que ofereix la capacitat de mesurar deformacions superiors a les 10.000 microdeformacions.

Les premisses del disseny han estat complir amb els requeriments dels usuaris i oferir una solució estable, robusta i fàcil de fer servir.

El projecte consta de dues parts diferenciades, el maquinari, encarregat de la adequació, amplificació i digitalització les senyals, i el programari que tracta i mostra les dades perquè les interpreti l'usuari.

Tot el projecte s'ha desenvolupat pels Serveis Tècnics de Laboratori a l'EPSEVG, i s'han produït 12 dispositius per fer-los servir a les pràctiques docents que es realitzen a la carrera de Enginyeria Tècnica Industrial Mecànica.

Paraules clau:

Extensiomètria, Deformació, Galga, Microcontrolador, PIC, Pont de Wheatstone, "*fully differential amplifier*", Instrumentació, USB, LabView, Convertidor Analògic Digital

Index

Introducció	7
Descripció del sistema	8

Maquinari

Introducció	10
Pont de Wheatstone	12
Amplificador d'instrumentació	15
Convertidor analògic – digital	17
Microcontrolador	19
Adaptador sèrie – USB	20
Layouts i llistat de components	21
Manual d'usuari	23

Programari PDUSADGE V8

Introducció	25
Funcionament	25
Programació i codi font	30

Codi C PIC 16F873

Main.C	37
Configuració.C	40

Introducció

Serveis Tècnics és la unitat encarregada de gestionar el laboratori de mecànica on el departament de resistència de materials de l'EPSEVG realitza pràctiques docents per a les titulacions d'Enginyeria Tècnica Industrial especialitat Mecànica on s'han de mesurar la elasticitat i resistència de diferents provetes.

El sistema de mesura que es va fer servir durant anys, era un sistema realitzat a mida per l'empresa CM4. Al canviar els ordinadors, l'antic sistema no es podia utilitzar per problemes de compatibilitat de maquinari. Amb aquesta problemàtica, els Serveis Tècnics de l'EPSEVG van desenvolupar un primer sistema de mesura de deformacions que ha estat en funcionament durant dos anys amb un rendiment satisfactori, però es van detectar unes limitacions que no permetien realitzar certes mesures amb unes condicions més exigents.

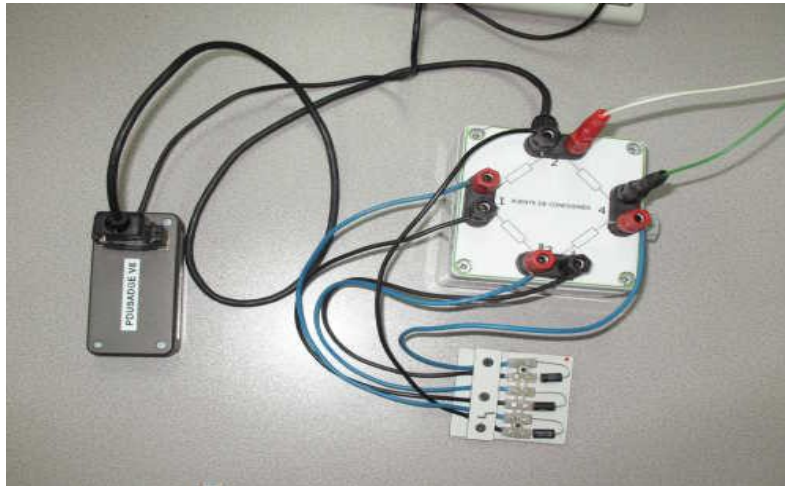
Per solucionar les mancances detectades, es va decidir desenvolupar un nou sistema amb unes prestacions superiors que permetés realitzar mesures amb totes les condicions presents a les pràctiques i a les futures modificacions de les mateixes.

El sistema que s'ha dissenyat respon a les peticions dels professors usuaris. Les especificacions bàsiques eren la d'un sistema amb una gran resolució, ràpid, estable, robust, sense limitacions alhora de canviar el maquinari i fàcil de fer servir pels usuaris.

El projecte és pot dividir en dos parts, una part de maquinari que conté l'electrònica encarregada de mesurar la deformació que pateix una proveta i transformar-la en informació digital, i una part de programari on s'adquireixen les dades, es tracten, s'analitzen i es mostren a la pantalla de l'ordinador com microdeformacions, que és la unitat amb que es treballa amb mesures de deformacions i elasticitat.

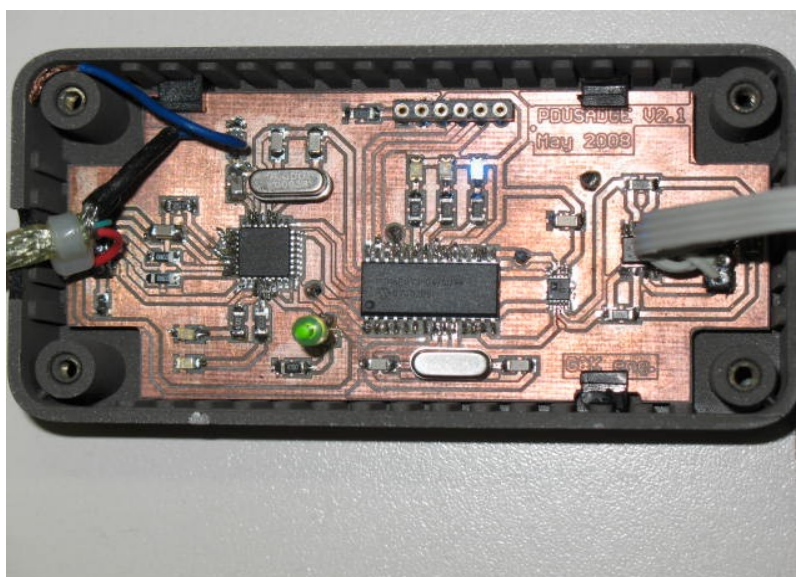
Descripció del sistema

Aquest dispositiu s'ha desenvolupat per a la docència de matèries de les carreres de Mecànica, per això un punt fonamental que s'ha tingut en compte alhora de dissenyar-lo ha estat la facilitat d'us per a l'usuari final i la robustesa del sistema.



El dispositiu finalment produït és una caixa amb apantallament que incorpora un connector DB15 (on es connecten els ponts de mesura) i un cable USB que s'ha de connectar a l'ordinador per la comunicació i l'alimentació.

Perquè el sistema funcioni, l'ordinador ha de disposar del programa de control PDUSADGE V8 i els arxius controladors del xip USB, que es poden descarregar de la pàgina web del fabricant.



La mesura de la deformació d'una proveta comença aplicant-l'hi una força que la deformarà i farà variar la resistència d'una galga adherida a la proveta. La galga es connecta en la configuració de pont de Wheatstone amb altres galgues o resistències de precisió.

El pont s'alimenta a 5V –mitjançant el mateix connector del sistema– i la seva sortida varia en funció de la deformació que pateix la proveta i la galga.

La tensió de sortida del pont s'aplica a l'amplificador d'entrada i sortida diferencial i que té un guany fixe $G=100$. Aquest senyal amplificat és el que s'ha de digitalitzar amb el convertidor analògic – digital.

La freqüència normal a la que es fa la conversió és de 100Hz, de manera que s'obtenen 100 mostres cada segon. Aquestes mostres son rebudes pel microcontrolador que les envia al controlador USB –quan rep l'ordre d'enviar, provinent del programa de control instal·lat a l'ordinador– finalment són enviades a l'ordinador per bus USB i es tracten al programa –PDUSADGE V8– que realitza un control d'errors, un filtratge i les converteix a les unitats de mesura de les pràctiques, les microdeformacions.

Maquinari

Introducció

La instrumentació és la part de la electrònica que s'encarrega del disseny d'instruments de mesura. El sistema que s'ha dissenyat es un dispositiu d'instrumentació que mesura la variació de la resistència de una galga extensiomètrica. Les galgues són resistències que varien el seu valor en funció de la variació de la seva longitud. Aquesta variació de resistència és proporcional a la deformació de tota la proveta, i quantificar-ho és l'objectiu de tot el sistema.

Les galgues extensiomètriques varien molt poc la seva resistència és per això que es fa servir una configuració anomenada "Pont de Wheatstone". Aquesta configuració permet mesurar variacions molt petites de resistències, la sortida del pont és directament un valor de tensió diferencial.

La tensió de sortida del pont de Wheatstone, encara és molt petita, per això s'ha d'amplificar. Per reduir al màxim la influència de la tensió d'alimentació, s'ha fet servir un amplificador diferencial, amb guany fixat en funció de les proves empíriques.

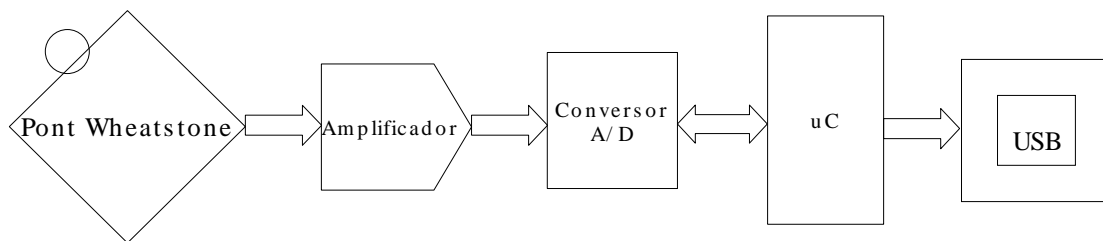
Un cop amplificada la sortida del pont, s'ha de digitalitzar per poder enviar-ho a l'ordinador. Es fa servir un convertidor analògic digital de 16 bits, que dona aproximadament 65000 punts de resolució. El convertidor també disposa de guany programable $\times 1$ $\times 2$ $\times 4$ i $\times 8$, i és comunica amb el microcontrolador mitjançant el protocol SPI.

El microcontrolador és l'encarregat de gestionar el funcionament de tot el sistema electrònic, té incorporat un protocol dissenyat a tal efecte per comunicar-se amb l'ordinador. Principalment engega i atura la transmissió d'informació cap a l'ordinador i varia el guany programable. Les dades digitals són transmeses via sèrie a un convertidor de nivells sèrie - USB.

El convertidor de nivells adapta les senyals digitals perquè puguin ser transmeses amb el protocol USB cap a l'ordinador de manera senzilla.

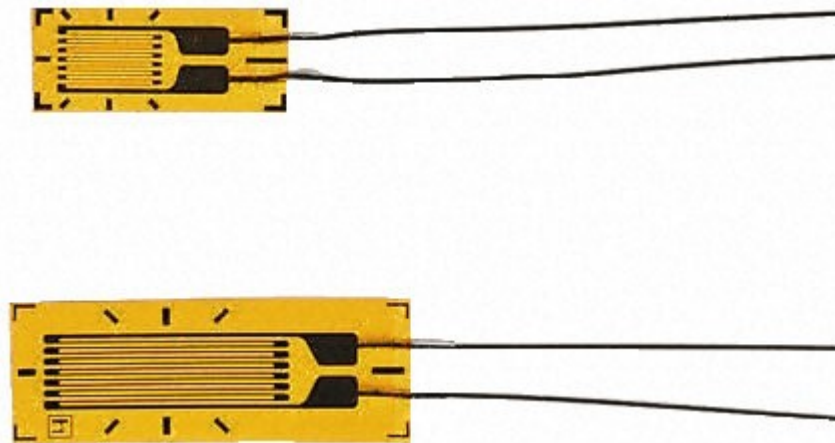
Tot el sistema rep l'alimentació del connector USB, de manera que no són necessàries fonts d'alimentació externes ni bateries. El dispositiu es posa en marxa automàticament al connectar-lo a l'ordinador.

L'esquema de blocs del sistema és el següent:

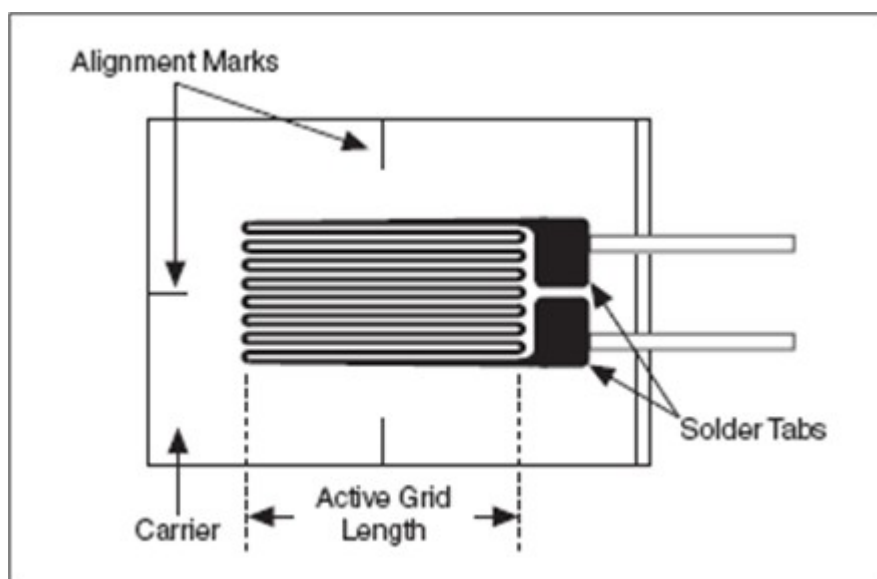


Pont de Wheatstone

El pont de Wheatstone és una configuració que es fa servir típicament per mesurar variacions molt petites d'elements resistius. El pont, típicament, està format per quatre resistències iguals, una o varies de les quals pot variar en funció d'algun paràmetre. En el cas que es presenta, el pont pot estar format per varies resistències i/o per galgues extensiomètriques.



Les galgues extensiomètriques són dispositius resistius amb una resistència nominal que varia en funció de la deformació que pateix. Altres factors, com la temperatura també influeixen en la seva resistència. Existeixen multitud de galgues en funció de la seva forma, mida, resistència nominal, tipus de material amb que estan fetes o materials sobre els que s'ha de muntar.

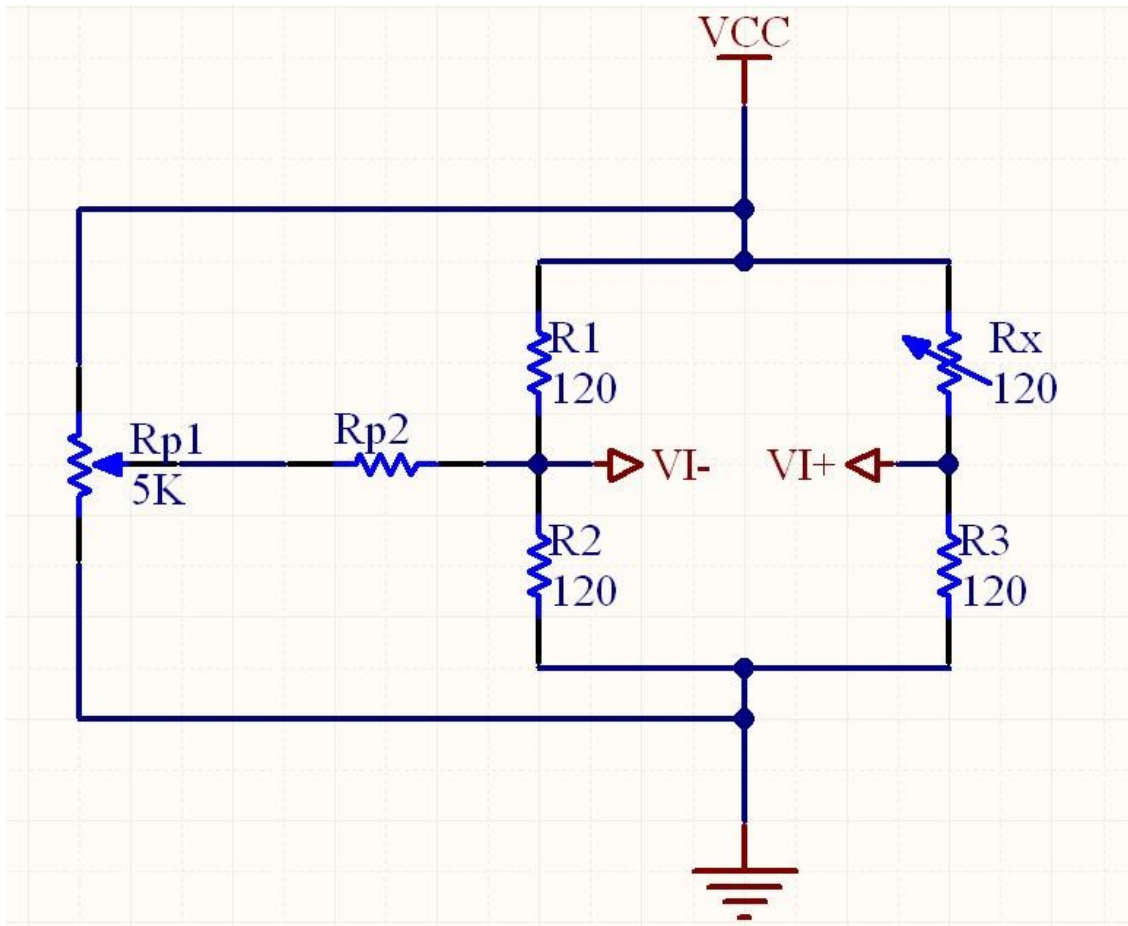


Per caracteritzar una galga, que s'anomena R_x , es fa servir la següent expressió:

$$R_x = R_0(1 + X)$$

On R_0 és el valor nominal de la galga. En aquest cas 120Ω .

L'esquema electrònic del pont de Wheatstone és el següent:



Cada branca del pont és com un divisor de tensió, la sortida de cada branca és:

$$VI- = \frac{VCC \cdot R2}{R1 + R2} \quad \text{i} \quad VI+ = \frac{VCC \cdot R3}{Rx + R3}$$

Essent totes les resistències iguals: $R1 = R2 = R3 = R$. La tensió diferencial VI és:

$$VI = VI+ - VI- = VCC \cdot \left(\frac{R}{R + Rx} - \frac{1}{R} \right)$$

En cas que el pont estigui en equilibri ($R_x=R$) s'obté:

$$V_I = V_{CC} \cdot \left(\frac{1}{R} - \frac{1}{R} \right) = 0$$

És a dir, quan el pont es troba en equilibri, la tensió entre les seves sortides (sortida diferencial) és zero, i cada sortida per separat, respecte a massa (GND), té una tensió de $V_{CC}/2$, ja que:

$$V_{I-} = \frac{V_{CC} \cdot R}{2R} = \frac{V_{CC}}{2} \quad V_{I+} = \frac{V_{CC} \cdot R}{2R} = \frac{V_{CC}}{2}$$

En el cas que ens ocupa hi han diferents configuracions possibles del pont de Wheatstone:

Una galga i tres resistències	Quart de pont
Dos galgues i dos resistències	Mig pont
Quatre galgues	Pont complert

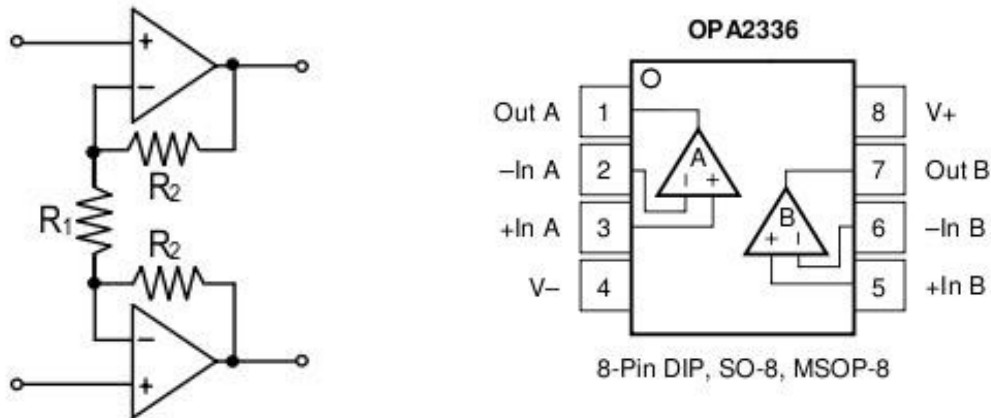
Depenent de quantes galgues i en quina configuració es munti el pont s'obtindrà una sortida o una altra per a una mateixa càrrega aplicada.

Els ponts de Wheatstone, que es fan servir a les pràctiques docents, tenen un potenciòmetre per ajustar l'equilibri del pont. Aquest sistema permet alterar l'equilibri del pont manualment i, d'aquesta manera, corregir petits desajustos que es puguin produir en el muntatge de les galgues.

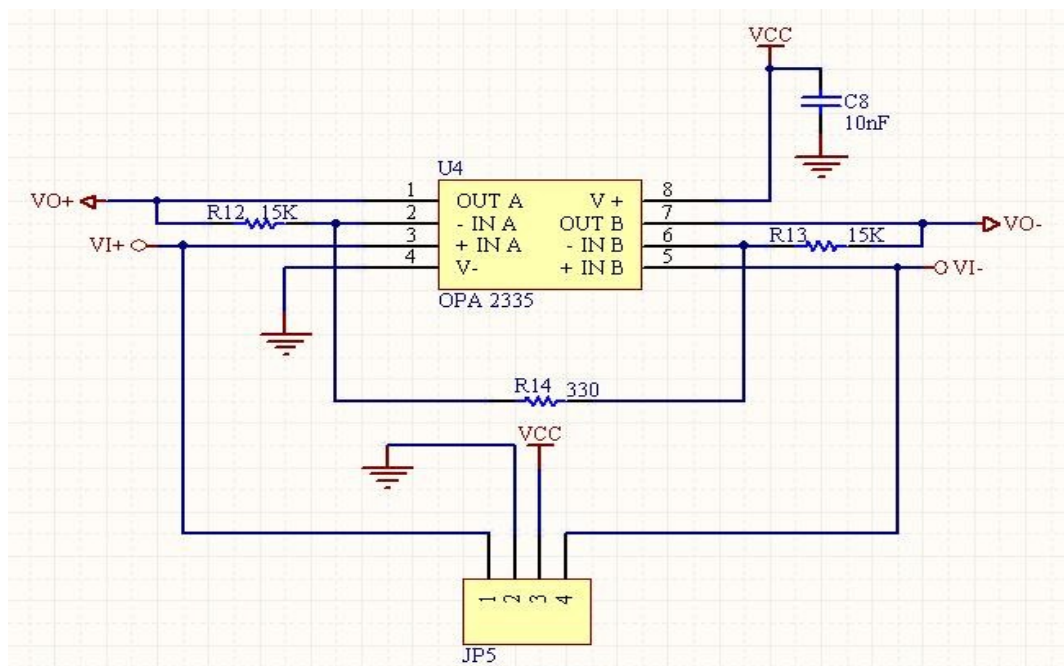
Mitjançant proves empíriques s'ha conegut quina podria ser la màxima tensió de sortida que pot existir en el pitjor cas, màxima diferència entre les branques del pont. D'aquesta manera s'ha configurat el guany de la següent etapa, l'amplificador d'instrumentació.

Amplificador d'instrumentació

El senyal de sortida del pont, és una tensió diferencial que s'ha d'amplificar per poder treballar amb el convertidor analògic - digital. Les característiques dels senyals amb que s'ha de treballar fan que s'hagi optat per una configuració completament diferencial, entrada diferencial i sortida també diferencial.



L'amplificador que s'ha fet servir és el OPA2336P de Burr Brown (Texas Instruments). Aquest integrat conté dos amplificadors operacionals que s'han connectat en configuració d'amplificador diferencial.



El guany de l'amplificador es pot variar actuant sobre la resistència R14 (Rg), les resistències R12 i R13 (R1) son iguals i de precisió. La funció de transferència de l'amplificador és:

$$\frac{V_o}{V_i} = \left(1 + \frac{2 \cdot R1}{R_g}\right)$$

La tensió diferencial de sortida del Pont de Wheatstone s'aplica a l'entrada de l'amplificador d'instrumentació i s'amplifica segons el guany configurat, en aquest cas el guany finalment és 100.

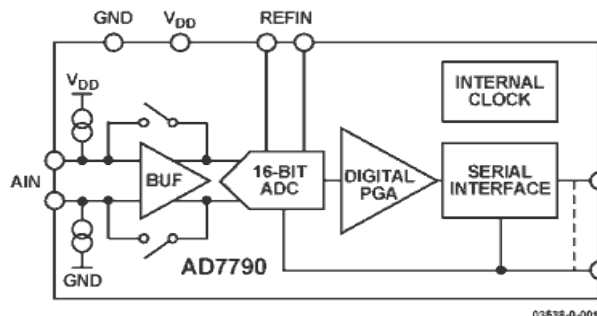
Aquesta configuració d'amplificador rep el nom de "*fully differential amplifier*" ja que tan l'entrada com la sortida són diferencials. S'ha escollit aquesta configuració per minimitzar els errors produïts per les variacions de la tensió d'alimentació. D'aquesta manera els petits canvis en la tensió d'alimentació, que es reproduïxen per tota la cadena de mesura, es compensen entre les dos branques diferencials tant a l'entrada com a la sortida.

Convertidor Analògic Digital

El senyal de sortida diferencial de l'amplificador s'ha de digitalitzar per poder treballar amb el microcontrolador i l'ordinador, per realitzar aquesta tasca s'ha optat per un convertidor A/D extern, ja que el convertidor que incorpora el microcontrolador només disposa de 10bits (1024 punts).

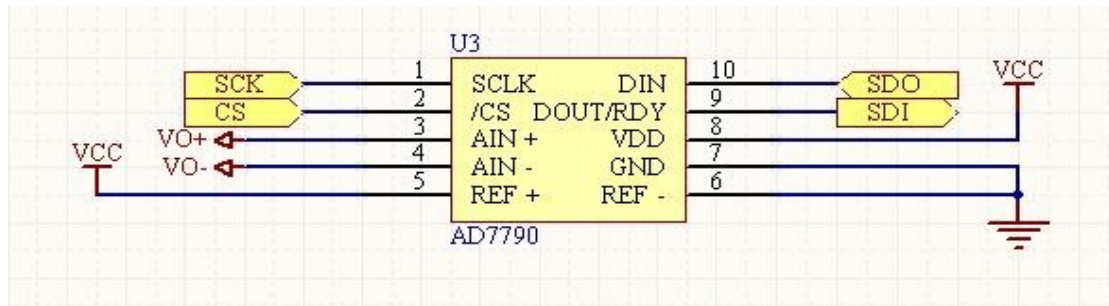
El convertidor escollit és el AD7790 de Analog Devices, disposa de 16 bits de resolució, entrada diferencial, guany programable i configuració de la velocitat de conversió també programable.

La comunicació es realitza mitjançant el protocol sèrie SPI. Aquest protocol que també implementa el microcontrolador, juntament amb la programació en llenguatge C, allibera al dissenyador de programar el control de les comunicacions entre aquests dos dispositius, ja que el microcontrolador actua de mestre i porta en tot moment el control del flux d'informació.



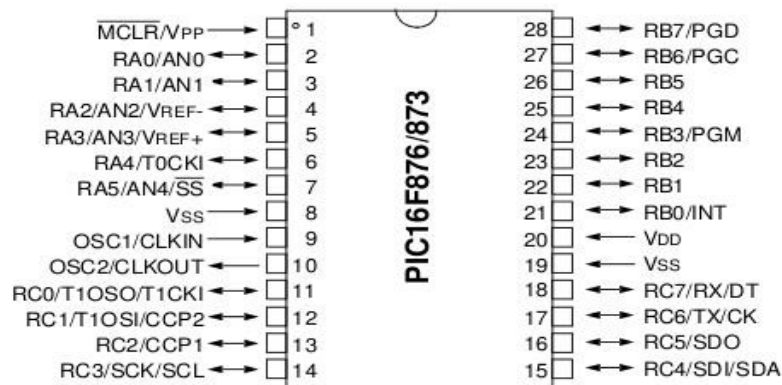
La tensió de referència del convertidor es pot configurar per l'usuari mitjançant un divisor de tensió, o es pot seleccionar directament la tensió d'alimentació. En aquest cas, s'ha seleccionat la tensió d'alimentació com a referència, que és igual a la tensió de cada una de les branques del pont en equilibri. Amb aquesta configuració, quan el pont es troba en equilibri –les dues branques tenen una tensió no diferencial igual a 2,5V– el convertidor dona una sortida digital igual a la meitat de tot el rang de sortida digital, aproximadament 32000 punts.

La resolució que s'ha aconseguit amb aquest sistema és suficientment gran, 16 bits equivalen a 65536 punts, si el marge de tensió mesurable és de 5V, la resolució que s'obté és de aproximadament 76,29uV.



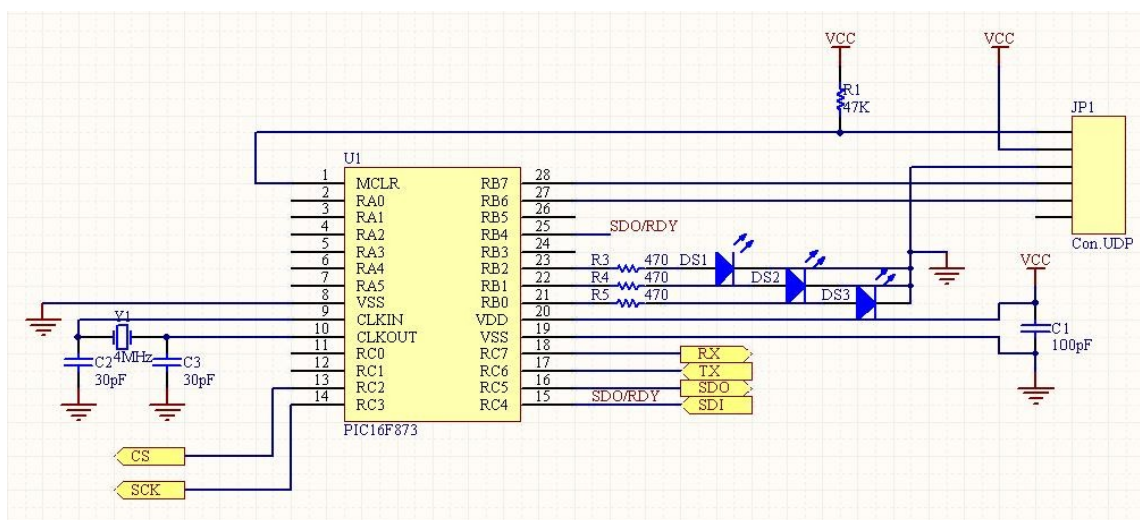
Microcontrolador

El microcontrolador és un PIC16F873, aquest dispositiu es fa servir principalment per controlar el protocol de comunicació amb l'ordinador, tractar les dades per poder transmetre-les pel bus USB i gestionar els diferents modes de treball del convertidor A/D.



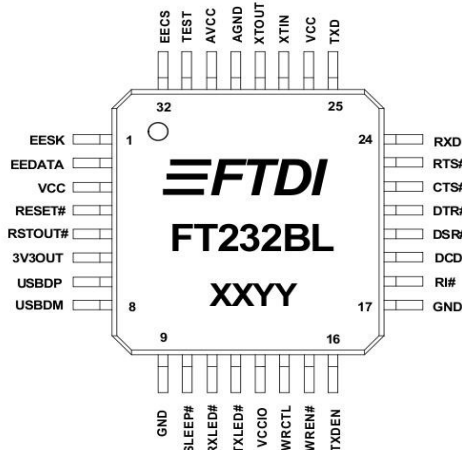
S'ha escollit aquest microcontrolador perquè té una estructura àmpliament coneguda, existeixen moltes eines de disseny disponibles i hi ha molta informació sobre el seu funcionament.

No s'han fet servir totes les possibilitats del microcontrolador, de fet, el circuit bàsic només fa servir les entrades del port SPI, la sortida sèrie i tres sortides digitals pels Led's de test i control.



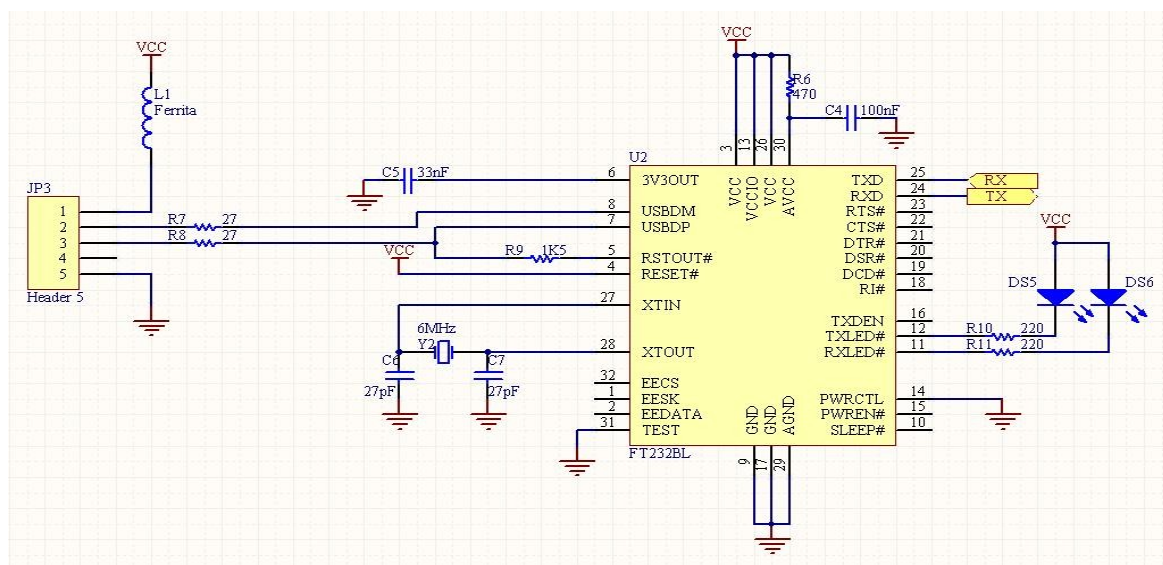
Adaptador Sèrie - USB

Tot el sistema s'alimenta i es comunica amb l'ordinador mitjançant el bus USB. S'ha escollit aquest tipus de comunicació per la seva popularitat i relativa facilitat d'ús que permet comunicar-se amb facilitat amb qualsevol ordinador i permet alimentar tot el circuit estalviant la implementació de fonts d'alimentació externes. Amb un sol cable es posa en marxa tot el dispositiu.



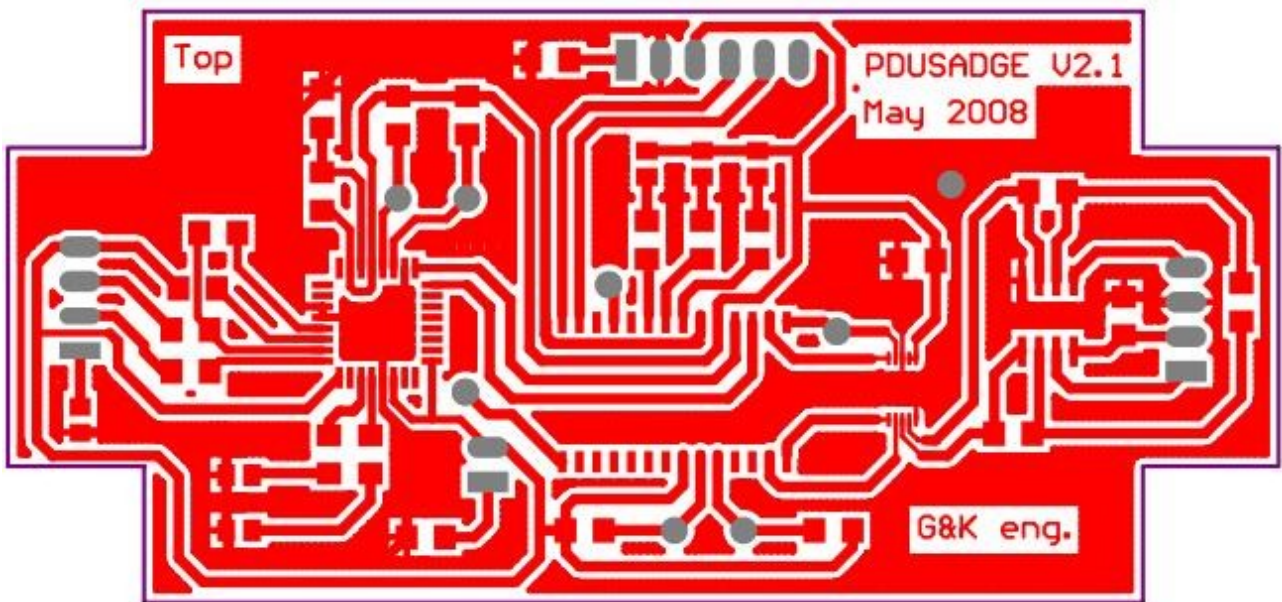
L'adaptador és un dispositiu discret que s'encarrega de gestionar tota la comunicació USB de manera que allibera al microcontrolador de gestionar el propi protocol USB. Aquest dispositiu incorpora un controlador per a l'ordinador que gestiona les dades i crea un port sèrie virtual tipus COMX de manera que l'aplicació d'usuari tracta les dades com si d'un port sèrie normal es tractés.

El dispositiu que s'ha fet servir és el FT232BL de la marca FTDI

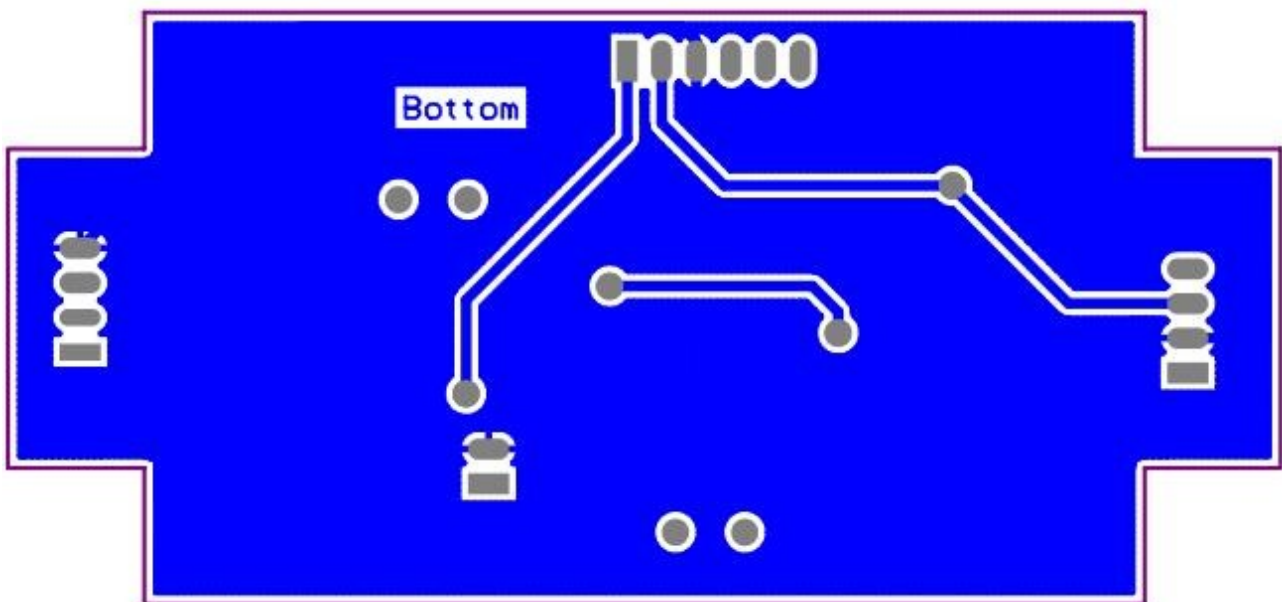


Layouts i llistat de components

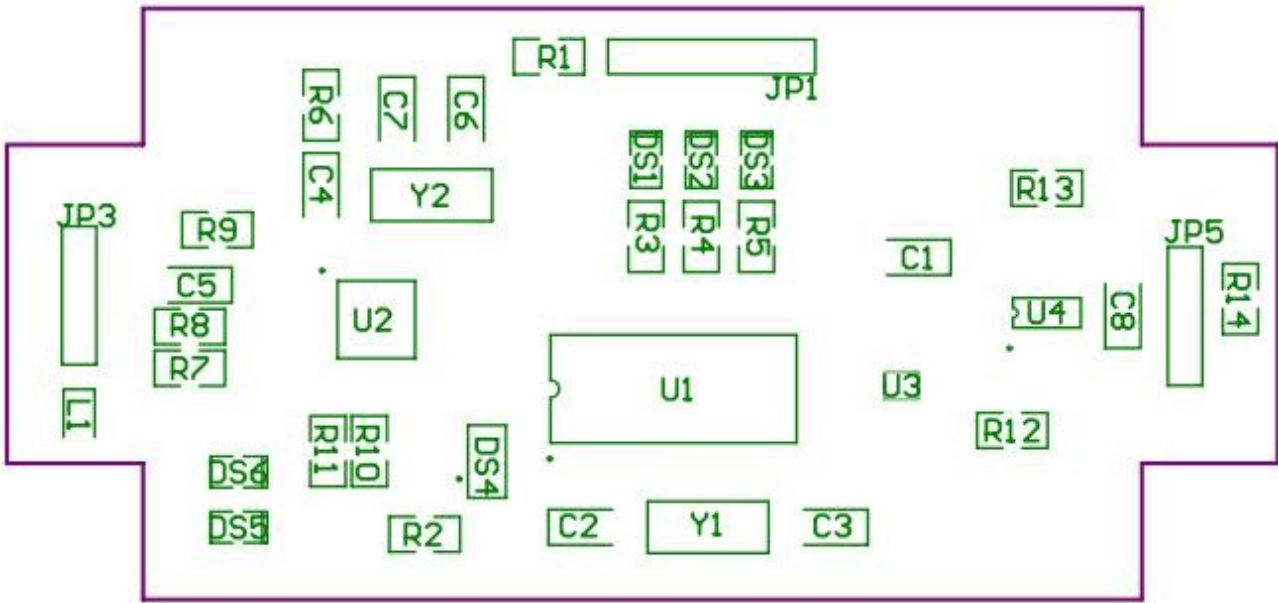
Top layer:



Bottom layer:



Components:



Llistat de components:

Comment	Designator	Value			
Cap	C1	100pF	Res1	R2	470
Cap	C2	30pF	Res1	R3	470
Cap	C3	30pF	Res1	R4	470
Cap	C4	100nF	Res1	R5	470
Cap	C5	33nF	Res1	R6	470
Cap	C6	27pF	Res1	R7	27
Cap	C7	27pF	Res1	R8	27
Cap	C8	10nF	Res1	R9	1K5
LED1	DS1		Res1	R10	220
LED1	DS2		Res1	R11	220
LED1	DS3		Res1	R12	15K
LED1	DS4		Res1	R13	15K
LED1	DS5		Res1	R14	330
LED1	DS6		PIC16F873	U1	
Con.UDP	JP1		FT232BL	U2	
USB	JP3		AD7790	U3	
Header 4	JP5		OPA 2335	U4	
Ferrita	L1	10mH	4MHz	Y1	
Res1	R1	47K	6MHz	Y2	

Manual d'usuari

L'usuari ha de fer el muntatge de la pràctica com correspongui segons l'enunciat. S'han de connectar les galgues a la caixa de connexions segons les indicacions del professor i es tanca el pont de Wheatstone amb les resistències de precisió que siguin necessàries.

Un cop el pont està muntat es pot connectar el dispositiu PDUSADGE a l'ordinador mitjançant un port USB lliure i s'engega el programa PDUSADGE V8. El programa detectarà el dispositiu connectat, llavors és quan l'usuari pot connectar el pont a la caixa amb el connector DB15. Quan el sistema detecti la caixa començarà a realitzar mesures en temps real.

En cas que la sortida del sistema no estigui suficientment equilibrada es pot actuar sobre el potenciòmetre de la caixa de connexions. Aquest potenciòmetre permet variar el punt d'equilibri del pont de Wheatstone manualment, d'aquesta manera es poden corregir petites variacions en el muntatge del pont produïts per desperfectes o problemes amb les galgues o la seva instal·lació.

Programari PDUSADGE V8

Introducció

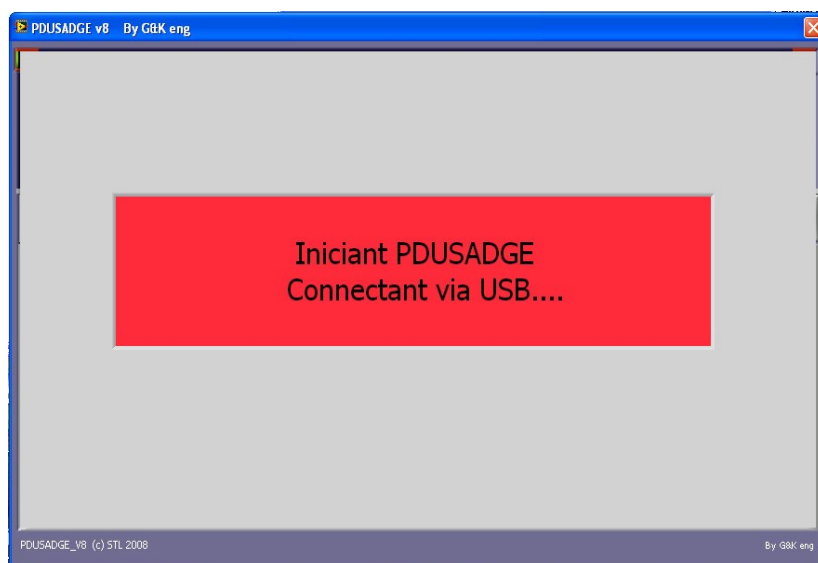
Un cop tenim les dades tractades en el maquinari hem de traslladar aquestes a un ordinador per ser mostrades a l'usuari d'una manera clara i ràpida. Per realitzar aquesta interfície d'usuari hem utilitzat el llenguatge de programació gràfica LabView.

Els objectius d'aquesta aplicació, que anomenarem PDUSADGE V8, són:

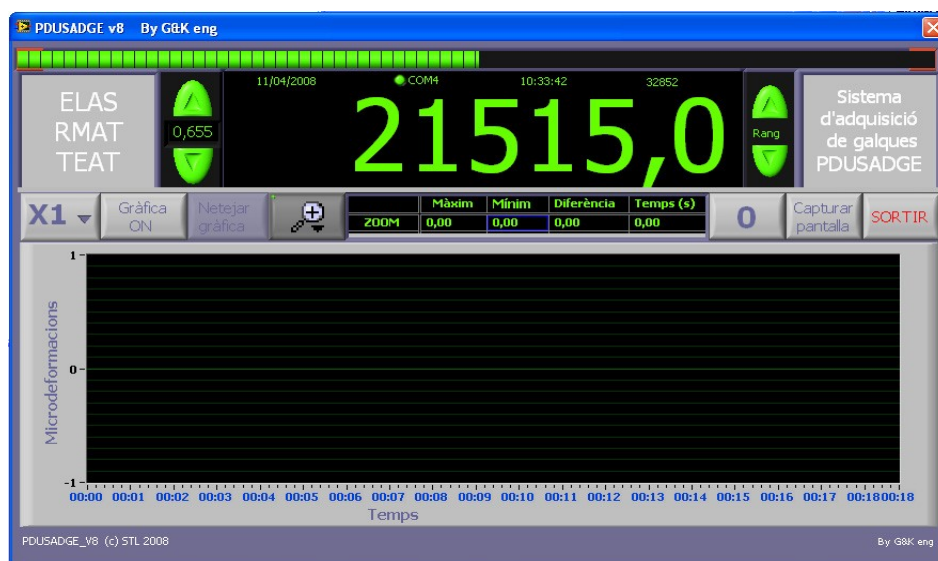
- Adquisició de dades mitjançant uns dels ports USB de l'ordinador
- Mostrar les dades de manera numèrica i en forma de gràfica
- Control per poder modificar i afinar el factor de calibració
- Poder variar el guany del PIC en el maquinari
- Capturar la pantalla en format jpeg

Funcionament

En el moment que s'inicia PDUSADGE V8, aquest fa una comprovació de tots els ports USB disponibles a l'ordinador per connectar amb el dispositiu que llegeix les dades del pont de Wheatstone. Fins que no detecta el dispositiu a la pantalla de l'ordinador ens apareixerà el següent:



Quan el programari detecti el dispositiu en un dels ports USB de l'ordinador, la pantalla anterior desapareixerà i es mostrarà la pantalla principal del PDUSADGE V8. Aquesta pantalla conté tots els elements necessaris per visualitzar i manipular les dades.



A la part superior de la pantalla observem una barra que ens indica la representació del desequilibri del pont. Aquesta barra trasllada el desequilibri del pont a punts de resolució del PIC del dispositiu. Va de 0 a 65536. En un pont ideal, la barra hauria d'estar exactament a la meitat.

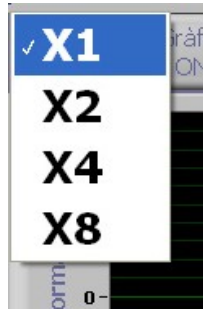
La representació numèrica la trobem en el gran display. En aquest display també podem veure a la part superior i en petit, la data, el port USB-COM en el que s'ha connectat el dispositiu, l'hora i les dades que rebem del dispositiu abans de ser tractades pel programa.

A la part esquerra del display tenim el factor de calibració que ens ha de permetre ajustar el valor necessari per la calibració del sistema abans de cada sessió de treball i/o després de variar el guany del dispositiu. Podem variar el valor introduint directament amb teclat o mitjançant les fletxes verdes.

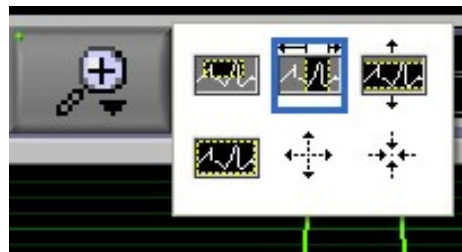
A la part dreta del display existeix un control que ens permet canviar el rang de la dada a representar tant al display com a la gràfica. Podem variar el rang mitjançant les fletxes verdes.

A la part central de la pantalla sobre la gràfica, tenim una sèrie de botons i una taula de dades del zoom.

- El primer botó X1 és un menú desplegable que ens permetrà canviar el guany del PIC del dispositiu. Tenim 4 opcions x1, x2, x4 o x8

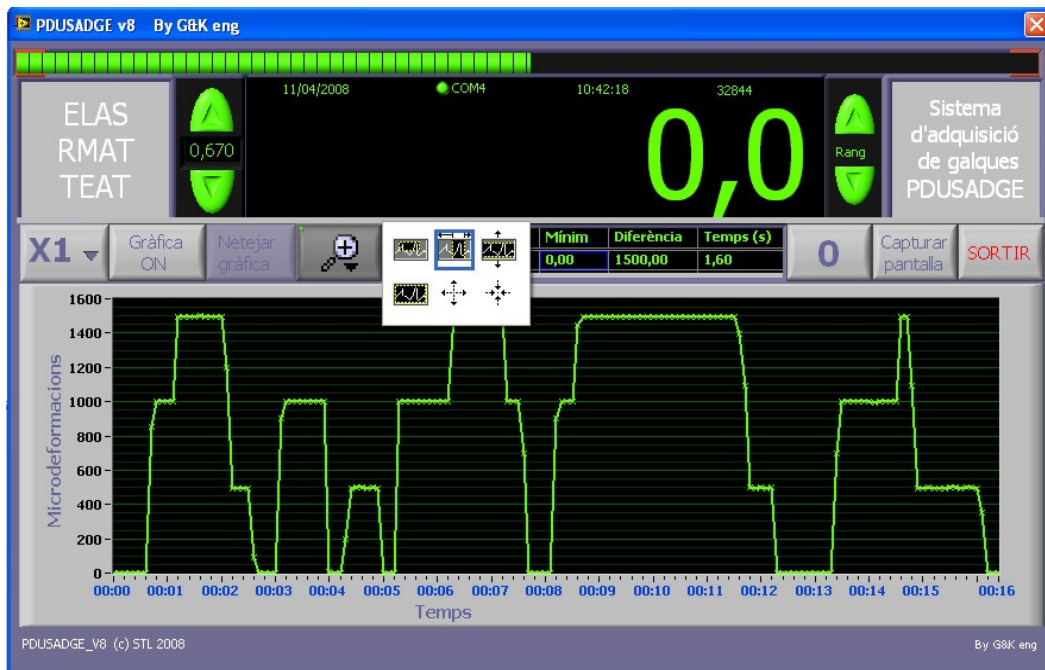


- El botó de Gràfica ON ens permet visualitzar o no la gràfica.
- El tercer botó ens permet netejar la gràfica.
- El botó de la lupa és la eina de zoom. Hi ha diferents tipus de zooms. Un cop escollit el zoom que volem utilitzar, anirem a la gràfica, farem el zoom i en la taula ens apareixeran els valors.

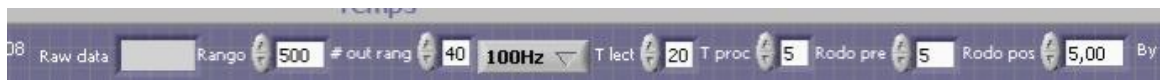


- El botó 0 fixa la referència en aquell moment a zero.
- El botó Capturar Pantalla ens captura en format jpeg tota la pantalla. Ens obre el Microsoft Paint amb la pantalla capturada i a partir d'aquí ja pots salvar la imatge jpeg.

La gràfica és la representació de les microdeformacions del pont en funció del temps.



Existeixen uns paràmetres de configuració ocults per a l'usuari. Un cop hem accedit a la part oculta ens apareixeran els paràmetres just sota la gràfica. I són:



D'esquerra a dreta tenim:

Raw data: Les dades que rebem del PIC sense tractar.

Rango: Valor+- en el que ha d'estar la propera dada que entri per donar-la per bona. És un filtre de dades aberrants.

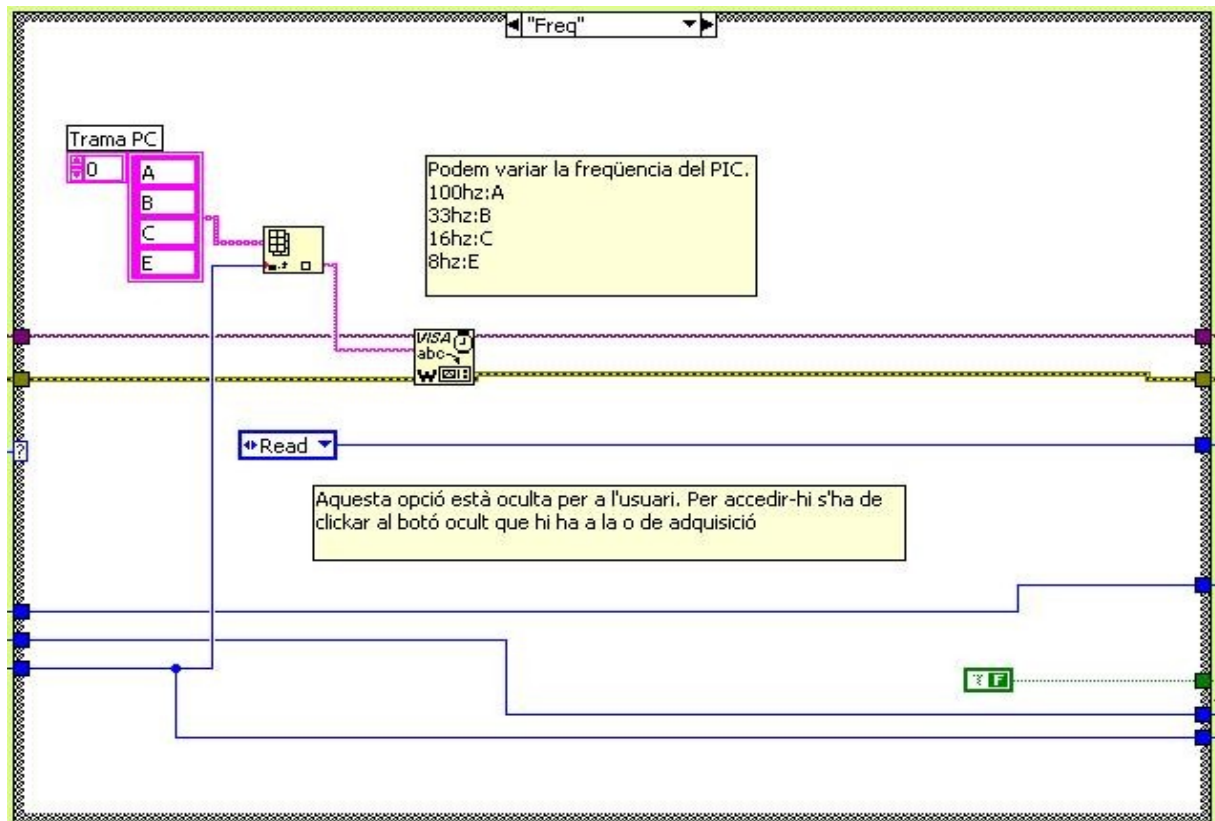
#out rang: Si durant 40 dades seguides rebudes no estem dintre de rang agafem la dada anterior que era bona.

100Hz: Menú desplegable en el que podem variar la freqüència del PIC.

Tlect: Temporitzador que tenim en el diagrama de blocs que fa la lectura de les dades del PIC.

Tproc: Temporitzador que tenim en el diagrama de blocs que fa el tractament de les dades un cop rebudes.

Rodopre i rodopos: Arrodoniment de les dades per tal de mostrar-les a l'usuari. Per defecte està a 5. Per tant les dades es mostraran en múltiples de 5.



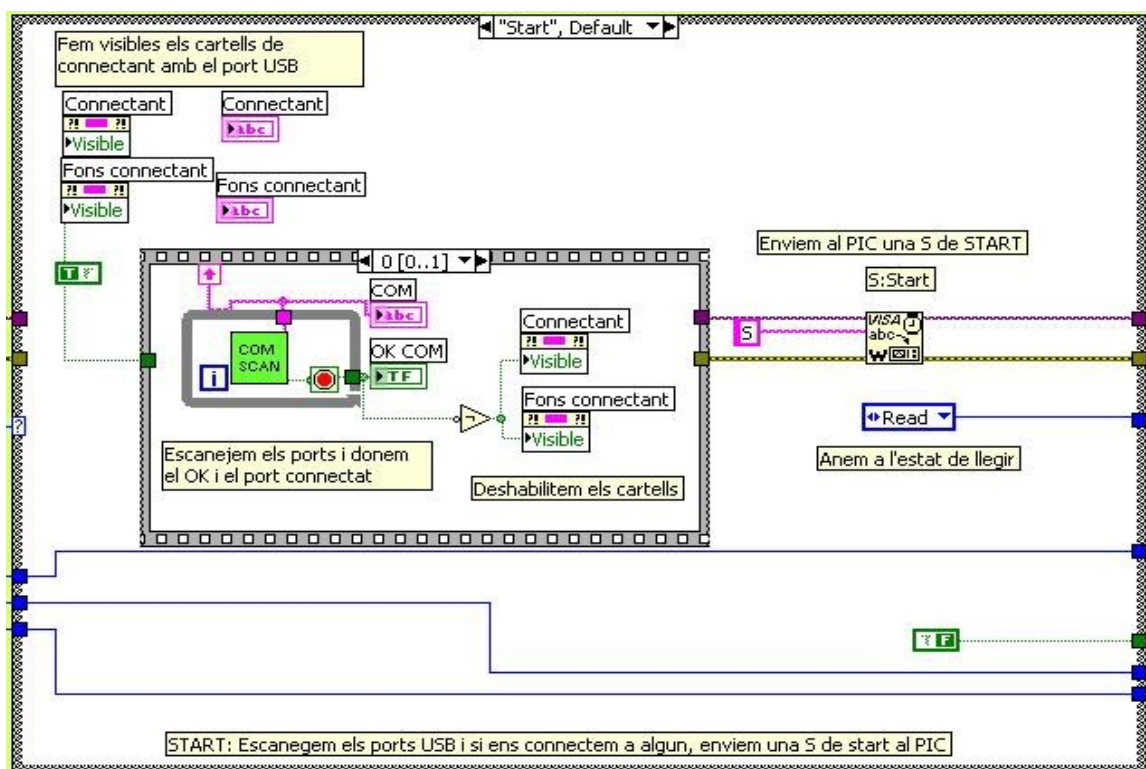
Programació i codi font

El codi font el podem dividir en tres blocs: Adquisició, Tractament i Representació de dades.

Adquisició de dades:

El bloc d'adquisició de dades és el de color verd. Aquest bloc està programat en el format de la màquina d'estats. Els estats dels que consta són: Start, Read, Mode, Mitjana, Freq i Stop

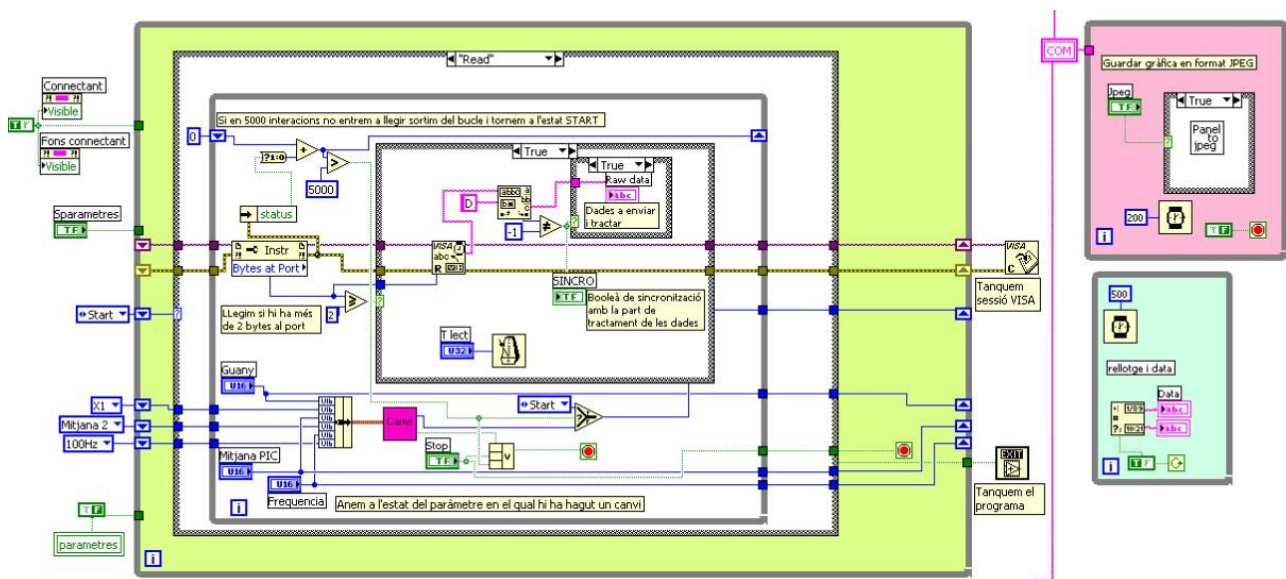
Start: La primera acció que fa és obrir una sessió VISA, escanejar tots els ports COM-USB de l'ordinador i si en un d'ells troba el PIC, activa la pantalla principal i envia una S al PIC per a que comenci a enviar. El bloc de START ens envia a READ.



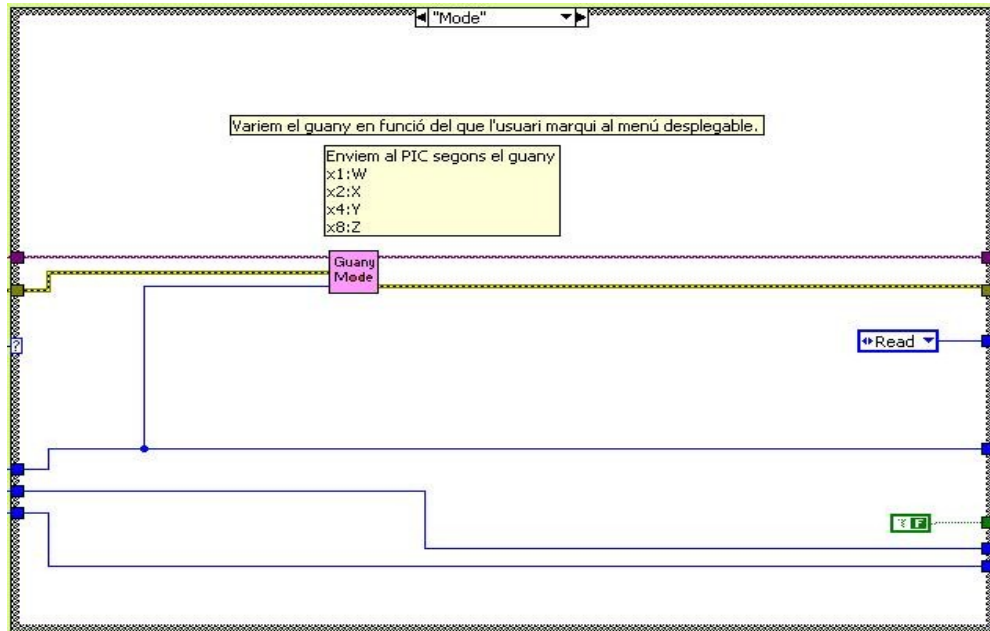
Read: Treballem sobre el port que ens envia l'estat Start. Llegim del buffer quan hi ha més de 2 bytes. Un cop tenim les dades extraïem la primera D, que és la marca, agafem la trama restant i l'enviem mitjançant una variable local, *Raw data*, les dades al bloc de Tractament de dades.

A la part superior de l'estat Read, tenim un time out que ens detecta possibles desconexions del cable o problemes de comunicació. Si en 5000 interaccions del bloc de lectura no llegim res, aquest ens envia novament a l'estat Start per tornar a establir la connexió.

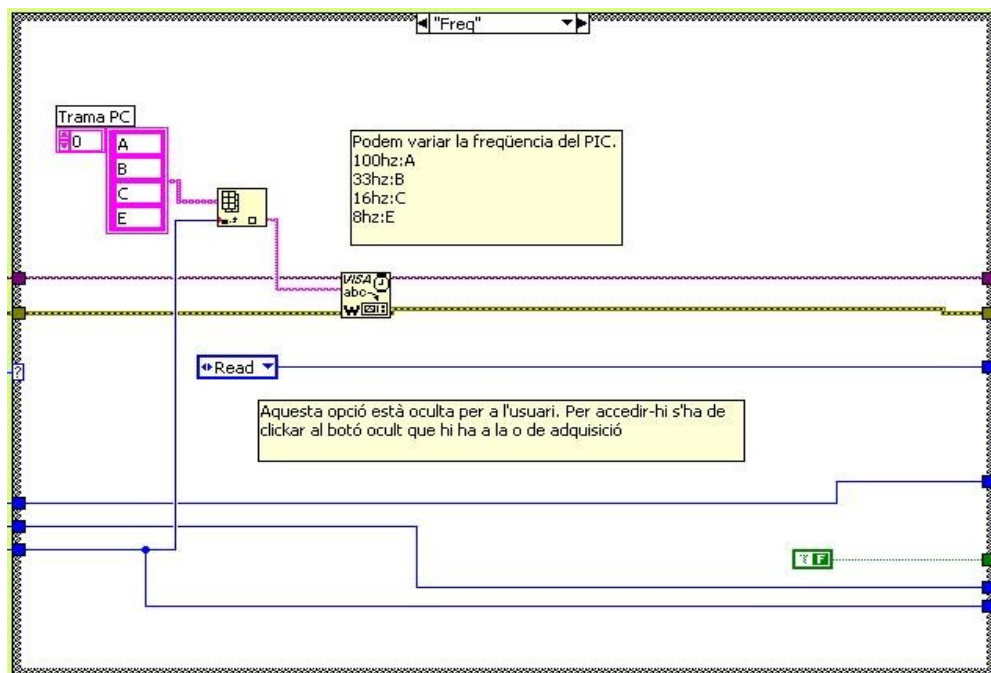
La part inferior de l'estat tenim la selecció dels diferents estats en funció de les modificacions dels controls per part de l'usuari. Els paràmetres que l'usuari pot modificar són: Mode, Freq i Mitjana. La sortida del bloc read per anar a un dels blocs de paràmetres es farà sempre que l'usuari modifiqui el valor del paràmetre. Comparem el valor actual amb l'anterior.



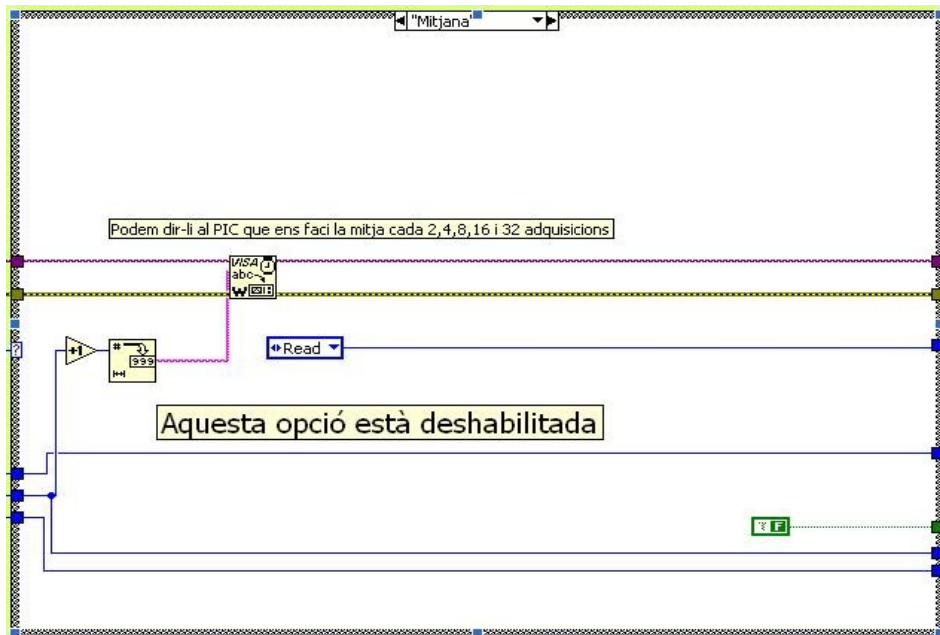
Mode: Aquest és l'estat en el que podem variar el guany del PIC. Segons l'elecció de l'usuari el guany el podem modificar en X1, X2, X4 i X8. Enviarem per a cada tipus de guany una lletra al PIC: W, X, Y, Z. Un cop enviada la trama per variar el guany tornarem a l'estat Read.



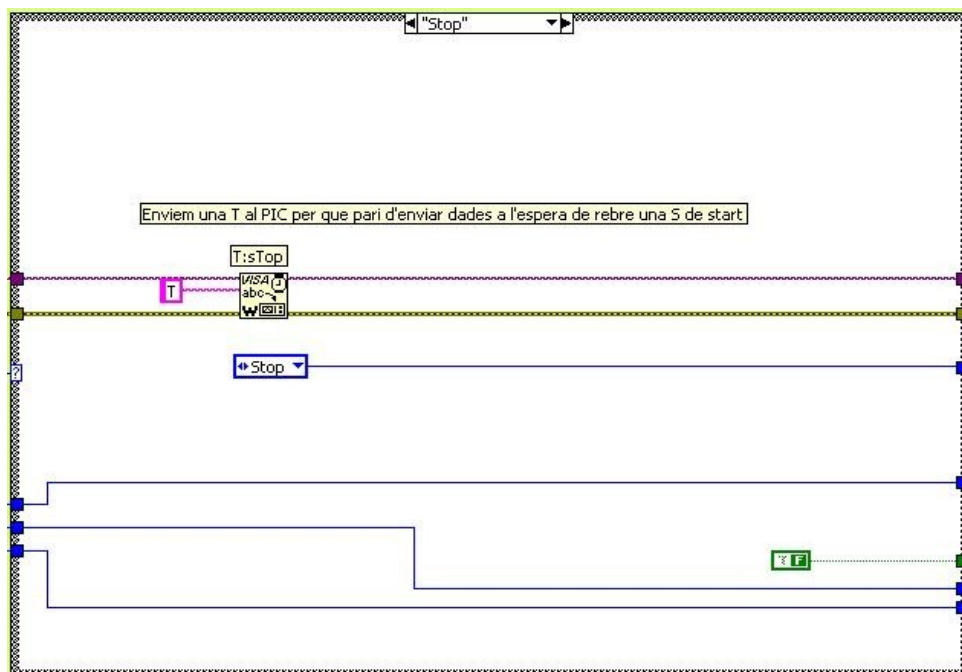
Freq: Variem la freqüència de conversió. Per cada freqüència enviem una lletra i tornem a Read. 100hz:A, 33hz:B, 16hz:C, 8hz:E.



Mitjana: Podem fer que el PIC ens faci una mitjana de les dades abans d'enviar-les per tenir menys variacions en l'adquisició. Actualment aquesta part està deshabilitada ja que donava molts problemes.



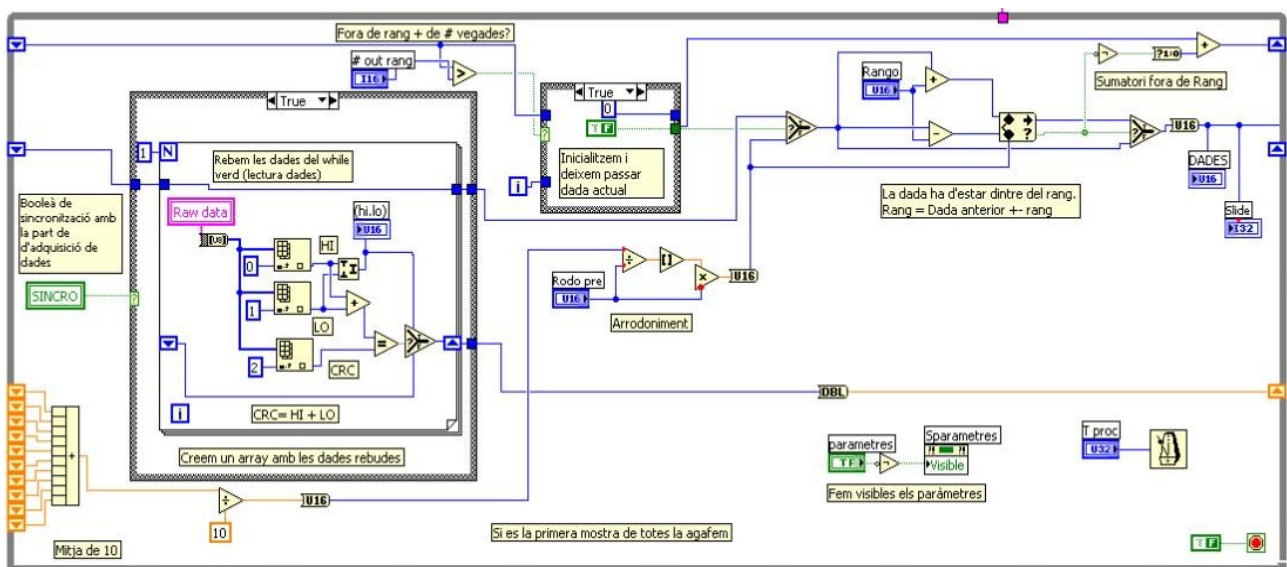
Stop: Enviem una T al PIC per a que aturi l'enviament de dades. Està programat però no està operatiu.



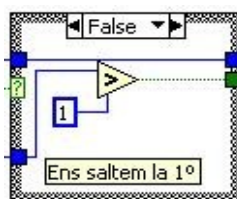
Tractament de dades:

Les dades són tractades per un bloc case structure que s'activa per sincronització amb el bloc d'adquisició Read (variable local *sincro*). Aquesta sincronització evita que repetim el tractament de les dades, ja que dins la variable local *Raw data* les dades quedarien en memòria.

Un cop rebem les dades, separem els 3 bytes que necessitem; Hi, Lo i CRC. El CRC és la suma de Hi i Lo. Unim Hi i Lo i obtenim la dada. Fem una mitja de 10 dades amb un Shift register. Posteriorment realitzem el primer arrodoniment. Després la dada la tractem per tal de mirar si està o no dintre del rang. Si està fora de rang, incrementem en 1 el sumatori del shift register. Si durant X vegades estem fora de rang, agafem la dada anterior vàlida i inicialitzem el sumatori de fora de rang.

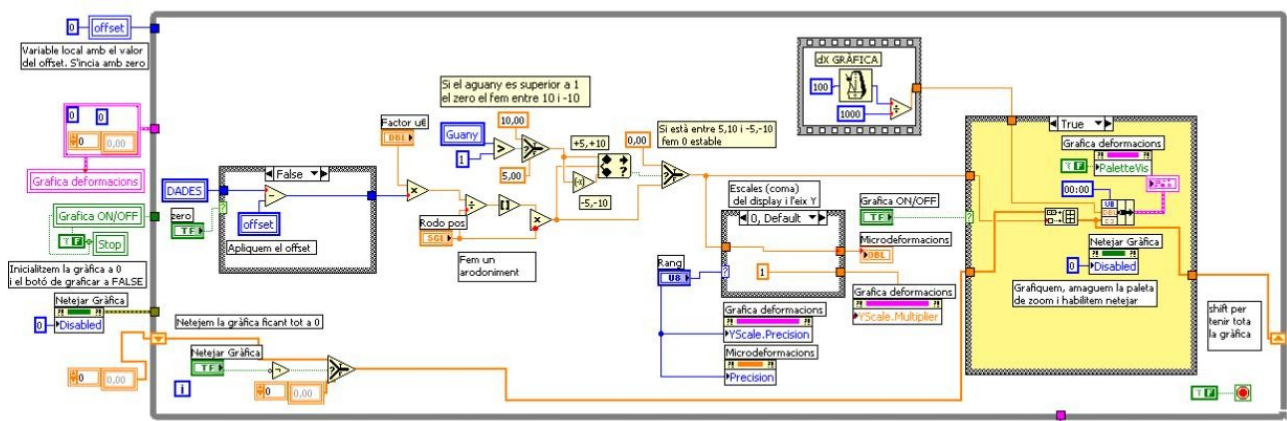


El case false evita el fora de rang ja que la primera dada no ha de passar cap filtre al no haver-hi dada anterior.



Representació de dades:

Les dades les rebem per la variable local DADES. Si l'usuari prem el botó ZERO, restem l'offset a la dada per aconseguir el zero virtual. Multipliquem pel factor i tornem a fer un altre arrodoniment. Per estabilitzar la senyal quan està a zero, fem zero quan les dades es troben entre $+5$ i -5 i $+10$ i -10 , depenent del guany seleccionant. Les dades les representem en una gràfica amb un bundle i la gràfica la anem acumulant en un shift register fins que l'usuari la neteja (fiquem array a 0). Fora del while inicialitzem totes les variables.



Codi C PIC 16F876

MAIN.C

```
//-----//  
/////////  
/////////  
PDUSADGE v2 (C) 2008  
//// SERVEIS TÈCNICS - ESCOLA POLITÈCNICA SUPERIOR D'ENGINYERIA DE VILANOVA I LA GELTRÚ  
//////// by GERMAN MORILLO, german.morillo@upc.edu  
/////////  
/////////  
//-----//  
#include <io16f873.h>  
#include "configuracio.h"  
#include <inpic.h>  
  
unsigned char  
dada_rebudaH,dada_rebudaL,txdata,rebuda,control,start,guany,dadaH,dadaL,media=8,frec,crc;  
unsigned long suma;  
int i=0,aux=0;  
  
#pragma vector=0x04 //El codi de la interrupció es fica a partir de la adreça 0X04  
__interrupt void interrupcion (void)  
{  
    GIE=0; //Deshabilito interrupcions per evitar colisions  
    if (RCIF==1) //Si s'ha rebut algo pel port sèrie, fes  
        RB0=1;  
    {  
        RCIF=0; //ficar interrupcions a 0 per poder rebre mes dades  
        rebuda=RCREG; //Llegeixo el registre d'entrada del port sèrie  
        if (rebuda=='P'){ //Si Rebo una P es porque s'està buscant el port sèrie  
            RB1=1;  
            enviar_datoint('K'); //Envio una K per confirmar que estic he rebut la P  
            control=1; //Activo control  
            RB2=1;  
        }  
        else if (rebuda=='W'){ //Guany conversor x1  
            enviar_datoint('W'); //Resposta rebut canvi configuració  
            enviar_datoint('Q');  
            guany=1; //Canvi variable guany  
            start=0; //Aturem bucle enviament  
        }  
        else if (rebuda=='X'){ //Guany conversor x2  
            enviar_datoint('X'); //Resposta rebut canvi configuració  
            enviar_datoint('Q');  
            guany=2; //Canvi variable guany  
            start=0; //Aturem bucle enviament  
        }  
        else if (rebuda=='Y'){ //Guany conversor x4  
            enviar_datoint('Y'); //Resposta rebut canvi configuració  
            enviar_datoint('Q');  
            guany=3; //Canvi variable guany  
            start=0; //Aturem bucle enviament  
        }  
        else if (rebuda=='Z'){ //Guany conversor x8  
            enviar_datoint('Z'); //Resposta rebut canvi configuració  
            enviar_datoint('Q');  
            guany=4; //Canvi variable guany  
            start=0; //Aturem bucle enviament  
        }  
        else if (rebuda=='S'){ //Start  
            start=1;  
        }  
        else if (rebuda=='T'){ //Stop  
            start=0;  
        }  
        else if (rebuda=='I'){ //Mitjana 2  
            media=2;  
        }  
    }  
}
```

```

    }
    else if (rebuda=='2'){           //Mitjana 4
        media=4;
    }
    else if (rebuda=='3'){           //Mitjana 8
        media=8;
    }
    else if (rebuda=='4'){           //Mitjana 16
        media=16;
    }
    else if (rebuda=='5'){           //Mitjana 32
        media=32;
    }
    else if (rebuda=='A'){           //Frecuencia 100Hz
        frec=0x01;
        start=0;
    }
    else if (rebuda=='B'){           //Frecuencia 33Hz
        frec=0x02;
        start=0;
    }
    else if (rebuda=='C'){           //Frecuencia 20Hz
        frec=0x03;
        start=0;
    }
    else if (rebuda=='E'){           //Frecuencia 16,6Hz
        frec=0x04;
        start=0;
    }
    }
    }
    GIE=1;                          //Activo interrupcions
}

void main (void)
{
    //Configuracio de les opcions del micro
    __set_configuration_word(XT_OSC & BODEN_OFF & DEBUG_OFF & CPD_OFF & LVP_OFF & CP_ALL &
    WRT_ENABLE_OFF & PWRTE_ON & WDT_OFF);
    configura_ports();
    configura_SPI();
    configura_lecturaSPI();
    configura_port_serie();
    configura_interrupcions();

    RB0=0;                          //Leds apagats
    RB1=0;
    RB2=0;
    start=0;                         //Inicialitzo bit de start
    control=0;                       //Inicialitzo el bit de control
    guany=1;                         //Inicialitzo el bit de guany
    RCIF=0;                          //Inicialitzo el flag de recepció pel port sèrie
    RCIE=1;                          //Habilito interrupció recepció port sèrie
    RC2=0;                           //Activo conversor A/D

    while(1){                        //Bucle principal
        while(control==0);           //Bucle espera configuració port sèrie
        while(start==1){             //Mentres no hi ha start no faig res
            while(RB4==1);           //Bucle principal de transmissió fa això mentre start=1
                spi_write(0x38);
                RB1=!RB1;
                RB2=!RB2;
                RB0=!RB0;
                dada_rebudaH=spi_read(txdata); //Llegeixo 8 bits de conversió pel SPI
                dada_rebudaL=spi_read(txdata); //Llegeixo els altres 8 bits
                dada_rebudaL=dada_rebudaL & 0xFC;
                crc=dada_rebudaL + dada_rebudaH;
                enviar_dato('D');
                enviar_dato(dada_rebudaH);    //Envio pel port sèrie 8 MSB
                enviar_dato(dada_rebudaL);    //Envio pel port sèrie 8 LSB
                enviar_dato(crc);
            }
        }

        switch(guany){
            case 1:

```

```

resetAD();
spi_write(0x10);          //Configuro guany x1
spi_write(0x02);
spi_write(0x20);          //Configuro frecuencia conversió 100Hz
spi_write(frec);
RB0=1;
RB1=0;
RB2=0;
start=1;
break;

case 2:
resetAD();
spi_write(0x10);          //Configuro guany x2
spi_write(0x12);
spi_write(0x20);          //Configuro frecuencia conversió 100Hz
spi_write(frec);
RB0=0;
RB1=1;
RB2=0;
start=1;
break;

case 3:
resetAD();
spi_write(0x10);          //Configuro guany x4
spi_write(0x22);
spi_write(0x20);          //Configuro frecuencia conversió 100Hz
spi_write(frec);
RB0=1;
RB1=1;
RB2=0;
start=1;
break;

case 4:
resetAD();
spi_write(0x10);          //Configuro guany x8
spi_write(0x32);
spi_write(0x20);          //Configuro frecuencia conversió 100Hz
spi_write(frec);
RB0=0;
RB1=0;
RB2=1;
start=1;
break;
    }
}
}

```

CONFIGURACIO.C

```
#include <io16f873.h>
#include "configuracio.h"

void configura_ports (void)
{
    TRISB=0x10;    //inicialitzo els ports salida leds
    TRISC=0x91;    //RC7, RC4 entrades, RC6,RC5, RC3-0 sortides
    TRISA=0x00;    //PORT A SORTIDA DESHABILITO EL SLAVE MODE
    RC2=1; //Chip Selec del conversors A/D desabilitat
}

void configura_interrupcions(void)    //pag 131
{
    PEIE=1;        //Habilita interruptor de periferics
    GIE=1;         //Habilitem interrupcions
}

void configura_SPI(void)
{
    CKP=1;         //quan no fa res esta a 1.
    SMP=0;         //lectura al mig del rellotge.
    CKE=0;         //es transmeteix quan esta de baixa el SCK.
    SSPEN=1;       //habilita sck,sdo,sdi.
    SSPM3=0;       //velocitat,la mes baixa.
    SSPM2=0;       //Tosc=T/64.
    SSPM1=1;
    SSPM0=0;
}

void configura_lecturaSPI(void)    //pg 67
{
    SSPOV=0;       //NO overflow
    WCOL=0;        //NO colision
}

void enviar_dato(unsigned char outdata)
{
    while (!TXIF);    //bit de transmetir
    TXREG=outdata;    //registro q transmite
}

void enviar_datoint(unsigned char outdata2)
{
    while (!TXIF);    //bit de transmetir
    TXREG=outdata2;   //registro q transmite
}

void leer_dato(unsigned char indata)
{
    while (!RCIF);
    indata=RCREG;
}

unsigned char spi_read(unsigned char txdata)
{
    SSPBUF=txdata;
    while(!BF);
    return SSPBUF;
}

void spi_write(unsigned char rxdata)
{
    SSPBUF=rxdata;
    while(!BF);
}
```



```

void configura_port_sèrie(void)
{
    BRGH=1;
    SPBRG=25;           //velocitat per 9600
    TX9=0;              //Mode 8 bits TX
    TXEN=1;             //Habilitat transmissió
    SYNC=0;            //Mode asincron
    SPEN=1;            //Habilitat port sèrie
    RX9=0;             //Mode 8 bits RX
    CREN=1;
}

void resetAD(void)
{
    spi_write(0xFF);    //4 vegades 0xFF :Reset del A/D
    spi_write(0xFF);
    spi_write(0xFF);
    spi_write(0xFF);
}

```